



TANGO Device Server

Booster sextupoles control User's Guide

Sextupoles Class

Revision: - Author: perez
Implemented in C++ - CVS repository: ESRF

Introduction:

Properties:

Device Properties		
Property name	**Property type**	**Description**
Pulse_delay	Tango::DEV_USHORT	moving pulse length
Deadband_delay	Tango::DEV_USHORT	alarm deadband delay
Alarm_gap	Tango::DEV_FLOAT	max difference between read and write on current RMS
Val_max	Tango::DEV_FLOAT	max value in amps of a wave point

Device Properties Default Values:

Property Name	Default Values
Pulse_delay	No default value
Deadband_delay	No default value
Alarm_gap	No default value
Val_max	No default value

There is no Class properties.

Commands:

More Details on commands....

Device Commands for Operator Level		
Command name	**Argument In**	**Argument Out**
Init	DEV_VOID	DEV_VOID
State	DEV_VOID	DEV_STATE
Status	DEV_VOID	CONST_DEV_STRING
DevStop	DEV_LONG	DEV_VOID
DevReset	DEV_LONG	DEV_VOID
DevStart	DEV_LONG	DEV_VOID
DevGetWave	DEV_LONG	DEVVAR_SHORTARRAY
DevSetWave	DEV_LONG	DEV_VOID
DevGetParams	DEV_LONG	DEVVAR_FLOATARRAY
DevSetParams	DEVVAR_FLOATARRAY	DEVVAR_FLOATARRAY
DevGetMeasures	DEV_LONG	DEVVAR_FLOATARRAY
DevGetWaveSize	DEV_VOID	DEV_LONG
DevChannelState	DEV_LONG	DEV_STATE
DevChannelStatus	DEV_LONG	DEV_STRING
DevReadSigValues	DEV_VOID	DEVVAR_FLOATARRAY
DevSetBpssT0	DEV_LONG	DEV_VOID
DevGetBpssT0	DEV_VOID	DEV_LONG

1 - Init

- **Description:** This commands re-initialise a device keeping the same network connection.
  After an Init command executed on a device, it is not necessary for client to re-connect to the device.
  This command first calls the device *delete_device()* method and then execute its *init_device()* method.
  For C++ device server, all the memory allocated in the *nit_device()* method must be freed in the *delete_device()* method.
  The language device desctructor automatically calls the *delete_device()* method.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  DEV_VOID : none.

- **Command allowed for:**

2 - State

- **Description:** This command gets the device state (stored in its *device_state* data member) and returns it to the caller.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  DEV_STATE : State Code

- **Command allowed for:**

3 - Status

- **Description:** This command gets the device status (stored in its *device_status* data member) and returns it to the caller.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  CONST_DEV_STRING : Status description

- **Command allowed for:**

4 - DevStop

- **Description:** Switch off the corresponding supply channel and stop the timing generation
- **Argin:**
  DEV_LONG : channel
- **Argout:**
  DEV_VOID :
- **Command allowed for:**

5 - DevReset

- **Description:** Reset the corresponding supply channel
- **Argin:**
  DEV_LONG : channel
- **Argout:**
  DEV_VOID :
- **Command allowed for:**

6 - DevStart

- **Description:** Switch on the corresponding supply channel and start the timing generation on the board
- **Argin:**
  DEV_LONG : channel
- **Argout:**
  DEV_VOID :
- **Command allowed for:**

7 - DevGetWave

- **Description:** Will read for the specified channel (first is 0) the contains of the Device Driver wave buffer (i.e. points acquired on the ADC for the specified channel). There are 2 DACs and each has 2 ADCs associated.
- **Argin:**
  DEV_LONG : channel

- **Argout:**
  DEVVAR_SHORTARRAY : current wave ADC buffer

- **Command allowed for:**

8 - DevSetWave

- **Description:** Set on the specified DAC channel (first is 0) the wave calculated using the current parameters (previously download).

- **Argin:**
  DEV_LONG : channel

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

9 - DevGetParams

- **Description:** Will return the current parameters used to calculate the wave on the specified DAC channel (first is 0

- **Argin:**
  DEV_LONG : channel

- **Argout:**
  DEVVAR_FLOATARRAY : wave calculation parameters

- **Command allowed for:**

10 - DevSetParams

- **Description:** Will update the wave calculation parameters and return a caculated wave (unit is amps) fot the specified DAC channel (first is 0) given as first array element.

- **Argin:**
  DEVVAR_FLOATARRAY : channel + wave caculation parameters

- **Argout:**
  DEVVAR_FLOATARRAY : wave points (amps)

- **Command allowed for:**

11 - DevGetMeasures

- **Description:** Will return the state and then the caculated DC and RMS of the voltage and current for the specified channel (first is 0). Note: calculation is done on ADC buffers.

- **Argin:**
  DEV_LONG : channel

- **Argout:**
  DEVVAR_FLOATARRAY : state, DC and RMS for U and I

- **Command allowed for:**

12 - DevGetWaveSize

- **Description:** Will return the number of points in a wave profile. This is a fixed value (typically 256).

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_LONG : wave number of points

- **Command allowed for:**

13 - DevChannelState

- **Description:** Will return the state of the corresponding channel (focusing or defocusing supply).

- **Argin:**
  DEV_LONG : channel

- **Argout:**
  DEV_STATE : channel state

- **Command allowed for:**

14 - DevChannelStatus

- **Description:** Will return the status of the corresponding channel (focusing or defocusing supply

- **Argin:**
  DEV_LONG : channel

- **Argout:**
  DEV_STRING : channel status

- **Command allowed for:**

15 - DevReadSigValues

- **Description:** Get for both channels the calculated DC and RMS of the voltage and current ADC buffers.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEVVAR_FLOATARRAY : DC_U RMS_U DC_I RMS_I

- **Command allowed for:**

16 - DevSetBpssT0

- **Description:** Force the T0 period used by the DD (valued changed by the BPSS DS and set through the Datacollector and a dedicated TANGO client

- **Argin:**
  DEV_LONG : T0 period in musec

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

17 - DevGetBpssT0

- **Description:** Get the T0 period currently used by the DD (measured at DS startup and then forced through DS command DevSetBpssT0)

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_LONG : T0 period in musec

- **Command allowed for:**

TANGO is an open source project hosted by : SOURCEFORGE.NET®	Core and Tools : CVS repository on tango-cs project Device Servers : CVS repository on tango-ds project